

UNIVERSITY OF MASSACHUSETTS
Dept. of Electrical & Computer Engineering

Introduction to Cryptography
ECE 597XX/697XX

Part 10

Digital Signatures

Israel Koren

ECE597/697 Koren Part.10 .1

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Content of this part

- ♦ The principle of digital signatures
- ♦ Security services
- ♦ The RSA digital signature scheme
- ♦ The Digital Signature Algorithm (DSA)
- ♦ The Elliptic Curve Signature Algorithm (ECDSA)

ECE597/697 Koren Part.10 .2

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

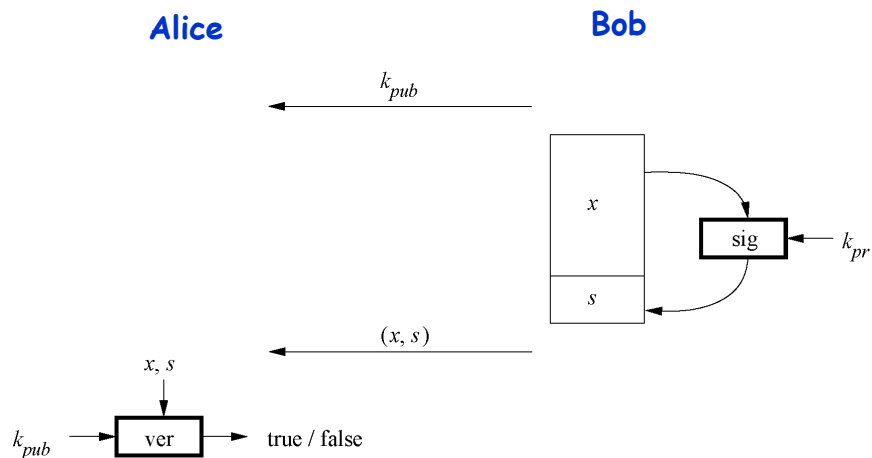
Motivation

- Bob orders a pink car from the car dealer Alice
 - After seeing the pink car, Bob states that he has never ordered it
 - How can Alice prove to a judge that Bob has ordered a pink car? (And that she did not fabricate the order herself)
- ⇒ Symmetric cryptography fails because both Alice and Bob can be malicious
- ⇒ Can be achieved with public-key cryptography

ECE597/697 Koren Part.10 .3

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Basic Principle of Digital Signatures



ECE597/697 Koren Part.10 .4

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Main idea

- For a given message x , a digital signature is appended to the message (just like a conventional signature).
 - Only the person with the private key should be able to generate the signature.
 - The signature must change for every document.
- ⇒ The signature is realized as a function with the message x and the private key as input.
- ⇒ The public key and the message x are the inputs to the verification function.

ECE597/697 Koren Part.10 .5

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Core Security Services

1. **Confidentiality**: Information is kept secret from all but authorized parties.
2. **Integrity**: Ensures that a message has not been modified in transit.
3. **Message Authentication**: Ensures that the sender of a message is authentic. An alternative term is data origin authentication.
4. **Non-repudiation**: Ensures that the sender of a message can not deny the creation of the message. (e.g., order of a pink car)

ECE597/697 Koren Part.10 .6

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Additional Security Services

5. **Identification/entity authentication:** Establishing and verification of the identity of an entity, e.g., a person, a computer, or a credit card.
6. **Access control:** Restricting access to the resources to privileged entities.
7. **Availability:** The electronic system is reliably available.
8. **Auditing:** Provides evidences about security relevant activities, e.g., by keeping logs about certain events.
9. **Physical security:** Providing protection against physical tampering and/or responses to physical tampering attempts
10. **Anonymity:** Providing protection against discovery and misuse of identity.

ECE597/697 Koren Part.10 .7

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Main idea of the RSA signature scheme

To generate the private and public key:

- Use the same key generation as RSA encryption.

To generate the signature:

- "encrypt" the message x with the private key

$$s = \text{sig}_{K_{\text{priv}}}(x) = x^d \bmod n$$

- Append s to message x

To verify the signature:

- "decrypt" the signature with the public key

$$x' = \text{ver}_{K_{\text{pub}}}(s) = s^e \bmod n$$

- If $x = x'$, the signature is valid

ECE597/697 Koren Part.10 .8

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

The RSA signature Protocol

Alice

Bob

← K_{pub}

$K_{pr} = d$
 $K_{pub} = (n, e)$

← (x, s)

Compute signature:

$s = \text{sig}_{K_{pr}}(x)$
 $\equiv x^d \bmod n$

Verify signature:

$x' \equiv s^e \bmod n$

If $x' \equiv x \bmod n \rightarrow$ valid signature*

If $x' \not\equiv x \bmod n \rightarrow$ invalid signature

• and the message was not modified in transit

The message x and signature s can be encrypted using AES

ECE597/697 Koren Part.10 .9

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Security and Performance of the RSA signature Scheme

Security:

The same constraints as RSA encryption: n needs to be at least 1024 bits to provide a security level of 80 bit.

⇒ The signature, consisting of s , needs to be at least 1024 bits long

Performance:

The signing process is an exponentiation with the private key and the verification process an exponentiation with the public key e .

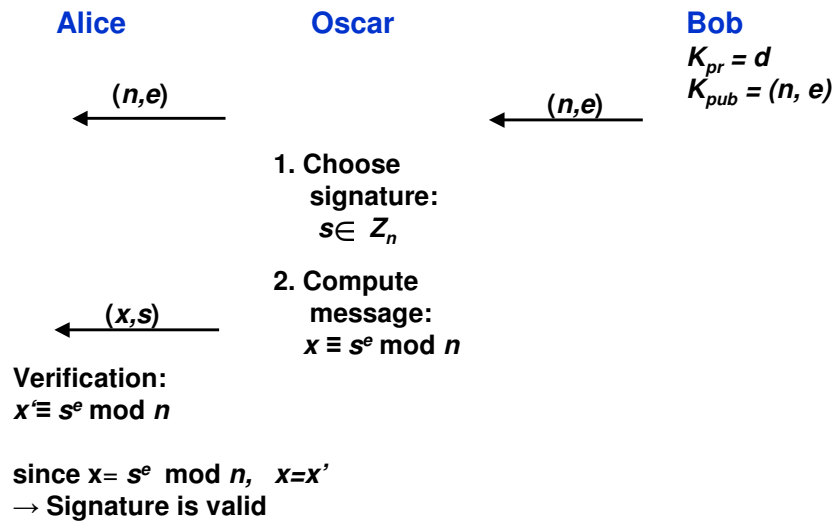
⇒ Signature verification is very efficient as a small number can be chosen for the public key.

Signed once but verified several times

ECE597/697 Koren Part.10 .10

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Existential Forgery Attack against RSA signature



ECE597/697 Koren Part.10 .11

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Existential Forgery and Padding

- An attacker can generate valid message-signature pairs (x,s)
- But an attack can only choose the signature s and NOT the message x

⇒ Attacker cannot generate messages like
„Transfer \$1000 into Oscar's account“

Formatting the message x according to a *padding scheme* can be used to make sure that an attacker cannot generate valid (x,s) pairs.

(A message x generated by an attacker during an Existential Forgery Attack will not coincide with the padding scheme.)

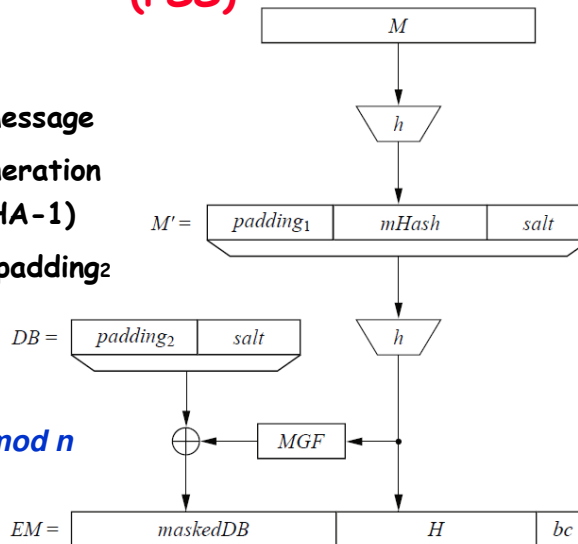
ECE597/697 Koren Part.10 .12

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

RSA Padding - Probabilistic Signature Standard (PSS)

- M - message
- EM - encoded message
- MGF - mask generation function (e.g., SHA-1)
- fixed $padding_1$, $padding_2$ and bc

$$\text{signature} = EM^d \bmod n$$



ECE597/697 Koren Part.10 .13

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Digital signature Algorithm (DSA) - background

- Federal US Government standard for digital signatures (DSS)
- Proposed by the National Institute of Standards and Technology (NIST)
- DSA is based on the ElGamal signature scheme
- Signature is only 320 bits long while the original ElGamal signature scheme requires 1024 or more
- Original ElGamal scheme is subject to an existential forgery attack
- Signature verification is slower compared to RSA

ECE597/697 Koren Part.10 .14

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Digital signature Algorithm (DSA) - 1

Key generation of DSA:

1. Generate a prime p with $2^{1023} < p < 2^{1024}$
2. Find a prime divisor q of $p-1$ with $2^{159} < q < 2^{160}$
3. Find an integer α with $\text{ord}(\alpha)=q$ (i.e., it generates the subgroup with q elements)
4. Choose a random integer d with $0 < d < q$
5. Compute $\beta \equiv \alpha^d \bmod p$

The keys are:

$$k_{pub} = (p, q, \alpha, \beta)$$

$$k_{pr} = (d)$$

ECE597/697 Koren Part.10 .15

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Digital signature Algorithm (DSA) - 2

DSA signature generation:

Given: message x , private key d and public key (p, q, α, β)

1. Choose a random (integer) ephemeral key k_E satisfying $0 < k_E < q$
2. Compute $r \equiv (\alpha^{k_E} \bmod p) \bmod q$
3. Compute $s \equiv (\text{SHA}(x) + d \cdot r) k_E^{-1} \bmod q$

The signature consists of (r, s)

SHA denotes the hash function SHA-1 which computes a 160-bit fingerprint of message x .

ECE597/697 Koren Part.10 .16

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Digital signature Algorithm (DSA) - 3

DSA signature verification:

Given: message x , signature (r, s) and public key (p, q, α, β)

1. Compute the inverse of s : $w \equiv s^{-1} \text{ mod } q$
2. Compute auxiliary value $u_1 \equiv w \cdot \text{SHA}(x) \text{ mod } q$
3. Compute auxiliary value $u_2 \equiv w \cdot r \text{ mod } q$
4. Compute $v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \text{ mod } p) \text{ mod } q$

If $v \equiv r \text{ mod } q \rightarrow$ signature is valid

If $v \not\equiv r \text{ mod } q \rightarrow$ signature is invalid

ECE597/697 Koren Part.10 .17

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Proof of DSA

We need to show that the signature (r, s) satisfies the condition

$$r \equiv v \text{ mod } q \text{ given } s \equiv (\text{SHA}(x) + d \cdot r) \cdot k_E^{-1} \text{ mod } q$$

$$\Leftrightarrow k_E \equiv s^{-1} \text{SHA}(x) + d \cdot s^{-1} r \text{ mod } q$$

$$\Leftrightarrow k_E \equiv u_1 + d \cdot u_2 \text{ mod } q$$

Raise α to either side of the equation and reduce modulo p :

$$\Leftrightarrow \alpha^{k_E} \text{ mod } p \equiv \alpha^{u_1 + d \cdot u_2} \text{ mod } p$$

Since $\beta \equiv \alpha^d \text{ mod } p$ we can write: $\alpha^{k_E} \text{ mod } p \equiv \alpha^{u_1} \beta^{u_2} \text{ mod } p$

We now reduce both sides of the equation modulo q :

$$\Leftrightarrow \underbrace{(\alpha^{k_E} \text{ mod } p) \text{ mod } q}_r \equiv \underbrace{(\alpha^{u_1} \beta^{u_2} \text{ mod } p) \text{ mod } q}_v$$

$$\Leftrightarrow r \equiv v$$

ECE597/697 Koren Part.10 .18

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Example

Alice

Bob

Key generation:

1. choose $p = 59$ and $q = 29$
2. choose $\alpha = 3$
3. choose private key $d = 7$
4. $\beta = \alpha^d = 3^7 \equiv 4 \pmod{59}$

$(p, q, \alpha, \beta) = (59, 29, 3, 4)$

Sign:

Compute hash of message

$H(x) = 26$

1. Choose ephemeral key $k_E = 10$
2. $k_E^{-1} = 3 \pmod{29}$
3. $r = (3^{10} \pmod{59}) \equiv 20 \pmod{29}$
4. $s = (26 + 7 \cdot 20) \cdot 3 \equiv 5 \pmod{29}$

$(x, (r, s)) = (x, (20, 5))$

Verify:

$H(x) = 26$

$w \equiv 5^{-1} \equiv 6 \pmod{29}$

$u_1 \equiv 6 \cdot 26 \equiv 11 \pmod{29}$

$u_2 \equiv 6 \cdot 20 \equiv 4 \pmod{29}$

$v = (3^{11} \cdot 4^4 \pmod{59}) \pmod{29} = 20$

$v \equiv r \pmod{29} \rightarrow$ valid signature

ECE597/697 Koren Part.10 .19

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Security of DSA

To solve the discrete logarithm problem in p the powerful index calculus method can be applied. But this method cannot be applied to the discrete logarithm problem of the subgroup q . Therefore q can be smaller than p .

p	q	hash output (min)	security levels
1024	160	160	80
2048	224	224	112
3072	256	256	128

Standardized parameter bit lengths and security levels for the DSA

ECE597/697 Koren Part.10 .20

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Use the ephemeral key only once

$$r \equiv (\alpha^{k_E} \bmod p) \bmod q = r_1 = r_2$$

$$s_1 \equiv (h(x_1) + d \cdot r) k_E^{-1} \bmod q$$

$$s_2 \equiv (h(x_2) + d \cdot r) k_E^{-1} \bmod q$$

If x_1 and x_2 are known

$$s_1 = (h(x_1) + d \cdot r) k_E^{-1} \bmod q \text{ and}$$

$$s_2 = (h(x_2) + d \cdot r) k_E^{-1} \bmod q$$

$$s_1 - s_2 = [h(x_1) - h(x_2)] k_E^{-1} \bmod q$$

Calculate k_E and then d

ECE597/697 Koren Part.10 .21

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Elliptic Curve Digital Signature Algorithm (ECDSA)

- Based on Elliptic Curve Cryptography (ECC)
- Bit lengths in the range of 160-256 bits can be chosen to provide security equivalent to 1024-3072 bit RSA (80-128 bit symmetric security level)
- One signature consists of two integers, hence the signature is twice the used bit length (i.e., 320-512 bits for 80-128 bit security level)
- The shorter bit length of ECDSA often result in shorter processing time

ECE597/697 Koren Part.10 .22

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Elliptic Curve Digital Signature Algorithm (ECDSA)

Key generation of ECDSA:

1. Use an elliptic curve with modulus p and coefficients a and b . Choose a point A that generates a cyclic group of primitive order q
2. Choose a random integer d with $0 < d < q$
3. Compute $B = dA$

The keys are:

$$k_{pub} = (p, a, b, q, A, B)$$

$$k_{pr} = (d)$$

ECE597/697 Koren Part.10 .23

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

ECDSA signature generation

ECDSA signature generation:

Given: message x , private key d and public key (p, a, b, q, A, B)

1. Choose a random (integer) ephemeral key k_E satisfying $0 < k_E < q$
2. Compute $R = k_E A$
3. Let r be the x -coordinate of R , x_R
4. Compute $s \equiv (h(x) + d \cdot r) k_E^{-1} \bmod q$

The signature consists of (r, s)

$h(x)$ is a hash function which computes a fingerprint of the message x .

ECE597/697 Koren Part.10 .24

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

ECDSA signature verification

ECDSA signature verification:

Given: message x , signature s and public key (p, a, b, q, A, B)

1. Compute the inverse of s : $w \equiv s^{-1} \bmod q$
2. Compute auxiliary value $u_1 \equiv w \cdot h(x) \bmod q$
3. Compute auxiliary value $u_2 \equiv w \cdot r \bmod q$
4. Compute $P = u_1 A + u_2 B$

If $x_P \equiv r \bmod q \rightarrow$ signature is valid

If $x_P \not\equiv r \bmod q \rightarrow$ signature is invalid

ECE597/697 Koren Part.10 .25

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Proof of ECDSA

We need to show that the signature (r, s) satisfies the condition

$$x_P \equiv r \bmod q \text{ given } s \equiv (h(x) + d \cdot r) \cdot k_E^{-1} \bmod q$$

$$\Leftrightarrow k_E \equiv s^{-1} h(x) + d \cdot s^{-1} r \bmod q$$

$$\Leftrightarrow k_E \equiv u_1 + d \cdot u_2 \bmod q$$

Multiply both sides of the equation by A

$$\Leftrightarrow \underbrace{k_E A}_{R} \equiv (u_1 + d \cdot u_2) A \bmod q = u_1 A + d \cdot u_2 A = \underbrace{u_1 A + u_2 B}_P$$

$$\Leftrightarrow r \text{ is the x-coordinate of } R, x_R$$

$$x_R = x_P$$

ECE597/697 Koren Part.10 .26

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Example

Alice

Bob

Key generation:

1. choose $p = 17$, and $a=b=2$
E: $y^2 = x^3 + 2x + 2$
2. choose $A=(5,1)$ with
 $q = \#E = 19$
3. choose $d = 7$
4. $B = d \cdot A = 7 \cdot (5,1) = (0,6)$

$$\xleftarrow{(p,a,b,q,A,B)=} (17,2,2,19,(5,1),(0,6))$$

Verify: $h(x)=26$ $\xleftarrow{(x,(r,s))=(x,(7,17))}$ Sign:

$w = 17^{-1} \equiv 9 \pmod{19}$
 $u_1 = 9 \cdot 26 \equiv 6 \pmod{19}$
 $u_2 = 9 \cdot 7 \equiv 6 \pmod{19}$
 $P = 6 \cdot (5,1) + 6 \cdot (0,6) = (7,11)$
 $x_P \equiv r \pmod{19} \rightarrow \text{valid signature}$

Compute hash of message

$h(x)=26$

1. Choose ephemeral key $k_E=10$
2. $R=10 \cdot (5,1)=(7,11)$
3. $r = x_R = 7$; $k_E^{-1} \equiv 2 \pmod{19}$
4. $s = (26 + 7 \cdot 7) \cdot 2 \equiv 17 \pmod{19}$

ECE597/697 Koren Part.10 .27

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources

Lessons Learned

- ♦ Digital signatures provide message integrity, message authentication and non-repudiation
- ♦ RSA is still the most widely used digital signature algorithm
- ♦ Competitors are the Digital Signature Standard (DSA) and the Elliptic Curve Digital Signature Standard (ECDSA)
- ♦ RSA verification can be done with short public keys e . Hence, RSA verification is usually faster than signing
- ♦ DSA and ECDSA have shorter signatures than RSA
- ♦ To prevent certain attacks, RSA should be used with padding
- ♦ The modulus of DSA and the RSA signature schemes should be at least 1024- bits long. For true long-term security, a modulus of length 3072 bits should be chosen. In contrast, ECDSA achieves the same security levels with bit lengths in the range 160-256 bits

ECE597/697 Koren Part.10 .28

Adapted from Paar & Pelzl, "Understanding Cryptography," and other sources